MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2019/20

ECE2216 MICROCONTROLLER AND MICROPROCESSOR SYSTEMS

(All sections / Groups)

15 OCTOBER 2019
9:00 a.m. – 11:00 a.m.
(2 Hours)

INSTRUCTIONS TO STUDENT

1. This Question paper consists of 10 pages including cover page and appendix with 4 Questions only.

2. Attempt **ALL** questions. All questions carry equal marks and the distribution of the marks for each question is given.

3. Please print all your answers in the Answer Booklet provided.

Question 1

(a) Given hexadecimal $A9_{16}$ and octal 64_8 are both **signed numbers** in their binary:

i. Convert both of them into **8-bit binary** format. Find their equivalent number in decimal. [5 marks]

ii. Add $A9_{16}$ to 64_8 using their **binary numbers** obtained from Q1(a)(i). Show your answer in binary. [3 marks]

iii. Identify the result obtained from Q1(a)(ii) as a positive or a negative number. [1 mark]

iv. Find the decimal equivalent value of the result obtained from Q1(a)(ii). Show the computation steps. [2 marks]

(b) What is the **on-chip capacity** of **ROM and RAM** of an 8051 microcontroller? [2 marks]

(c) What are the functions of Program Counter (PC), Instruction Register (IR), Arithmetic Logic Unit (ALU) and Data Register? [8 marks]

(d) A memory IC is specified as 1024K * 32bits. Evaluate the number of address lines and data lines required for this memory IC. [4 marks]

Continued ...

Question 2

(a) Explain the functions of the following 8051 control bus signals – Program Store Enable ($\overline{PSEN}$) and External Access ($\overline{EA}$) when they are used to select External memory.

[4 marks]

(b) What is Multiplexing? Illustrate how 1 K-bytes external RAM memory could be interfaced to the 8051 microcontroller on the arrangement of multiplexing address/data bus lines, and also use all control bus lines to complete the diagram. Justify the use of Address Latch Enable (ALE) pin of 8051.

[12 marks]

(c) Find out the PSW (Program Status Word) bits (i.e., carry flag, auxiliary carry flag, and parity bit flag) into accumulator affected values after executing the following instruction sequence of the codes:

```
MOV  R7, # 05FH
MOV   A, # 0F5H
ADD   A, R7
```

[6 marks]

(d) Determine the duration of a machine cycle if an 8051 microcontroller is operating with an 8 MHz crystal oscillator.

[3 marks]

Continued...

Question 3

(a) The incomplete instruction sequence and the machine codes for the 8051 microcontroller are shown in Table Q3(a).

Table Q3(a)

Address	Instruction	Machine codes
0000H	MOV A, #0B9H	
0002H		78H 25H
0004H	SUBB A, R0	
0005H		64H 5AH

i. Complete and reproduce the table in your answer script by converting each instruction to the corresponding machine codes and converting the machine codes to the corresponding assembly instructions. [4 marks]

ii. Identify the addressing mode of the source and destination operands for the first and third instructions. [2 marks]

iii. If a 12MHz crystal oscillator is used with the 8051 microcontroller, determine the total time for the microcontroller to execute this instruction sequence. [2 marks]

Hint: Refer to the opcode map in the Appendix.

(b) An embedded system engineer has developed an assembly program for the 8051 microcontroller as shown in Code Listing Q3(b):

```
        ORG 0000H
        MOV R0, #50H
        MOV R1, #02H
        MOV R2, #00H
        MOV R3, #03H
        MOV DPTR, #LOOKUP
LOOP:   MOV A, R1
        MOVC A, @A+DPTR
        MOV @R0, A
        INC R0
        INC R1
        ADD A, R2
        MOV R2, A
        DJNZ R3, LOOP
        SJMP $      ; the program stops here
LOOKUP: DB 01H, 04H, 09H, 10H, 19H, 24H
        END
```

Figure Q3(b)

Continued...

i. Analyze the program listed in Figure Q3(b) to determine the address of the affected memory locations in the RAM and their final contents when the program has stopped. [3 marks]

ii. Determine the final contents of registers R0, R1 and accumulator A when the program has stopped. [3 marks]

iii. Calculate the address represented by the label "LOOP". [2 marks]

Hint: Refer to the opcode map in the Appendix.

(c) Write an instruction sequence for the 8051 microcontroller to carry out the following arithmetic operation and store the result in register R3:

$$(14_{10} + 13_{10}) \times (61_{10} - 33_{10})$$

[4 marks]

(d) Develop an assembly program for the 8051 microcontroller to generate a 100Hz square wave (50% duty cycle) at pin P0.1 by using Timer 0 operating in Mode 1. Assume that the microcontroller is operating with a 12 MHz crystal oscillator. [5 marks]

Continued...

Question 4

(a) An 8051 microcontroller is required to **receive data** through serial port using 8-bit UART. Use Timer 1 with Mode 2, and baud rate of 2400 (with SMOD=0). The oscillator frequency of the microcontroller is 11.0592 MHz.

i. Fill in the required values for registers SCON and TMOD in the box below and reproduce them in your answer script. [3 marks]

SCON:	SM0	SM1	SM2	REN	TB8	RB8	TI	RI

TMOD:	GATE	C/T	M1	M0	GATE	C/T	M1	M0

ii. Determine the required value to be loaded into TH1 to set this baud rate. Show the calculation steps. [4 marks]

iii. Write the complete instruction sequence to receive 7 bit ASCII code. Expect odd parity in the 8th bit received and set the carry flag if there is a parity error. Then put the 7-bit ASCII in P1. [7 marks]

(b) The following instruction sequence is initialized by an 8051 microcontroller:

```
MOV IE, #9BH
MOV IP, #11H
```

i. Determine which interrupts have been enabled. [2 marks]

ii. Determine which interrupt has highest priority? [1 mark]

iii. Determine the sequence of servicing the interrupts if all enabled interrupts occur at the same time. [2 marks]

(c) An 8051 microcontroller is used to interface with two common-anode 7-segment LEDs. Port 1 is used to transfer data for display. Port P3.4 and P3.5 are used to control which LED is to turn on. Write the assembly program to display character "SO" to these two 7-segment LEDs.

[6 marks]

End of Paper

APPENDIX

Special Function Register Formats

Interrupt Enable (IE)

Bit Addr.	AFH	-	-	ACH	ABH	AAH	A9H	A8H
Name	**EA**	-	-	**ES**	**ET1**	**EX1**	**ET0**	**EX0**

BIT	SYMBOL	FUNCTION (Enable=1, Disable=0)
IE.7	EA	Global enable/disable. EA = 1, each individual source is enabled/disabled by setting/clearing its enable bit. EA = 0, disable all interrupts.
IE.6	-	Undefined
IE.5	-	Not implemented in 8051. ET2 for 8052.
IE.4	ES	Serial port interrupt enable bit.
IE.3	ET1	Timer 1 interrupt enable bit.
IE.2	EX1	External interrupt enable bit.
IE.1	ET0	Timer 0 interrupt enable bit.
IE.0	EX0	External interrupt enable bit.

Interrupt Priority (IP)

Bit Addr.	-	-	-	BCH	BBH	BAH	B9H	B8H
Name	-	-	-	**PS**	**PT1**	**PX1**	**PT0**	**PX0**

BIT	SYMBOL	FUNCTION (Enable=1, Disable=0)
IP.7	-	Undefined
IP.6	-	Undefined
IP.5	-	Not implemented in 8051. PT2 for 8052.
IP.4	PS	Serial port interrupt priority bit.
IP.3	PT1	Timer 1 interrupt priority bit.
IP.2	PX1	External interrupt priority bit.
IP.1	PT0	Timer 0 interrupt priority bit.
IP.0	PX0	External interrupt priority bit.

Interrupt Vectors

Interrupt Source	Flag	Vector Address
System Reset	RST	0000H
External 0	IE0	0003H
Timer 0	TF0	000BH
External 1	IE1	0013H
Timer 1	TF1	001BH
Serial Port	RI & TI	0023H
Timer 2 (8052)	TF2 or EXF2	002BH

Program Status Word (PSW)

Bit Addr.	D7H	D6H	D5H	D4H	D3H	D2H	-	D0H
Name	**CY**	**AC**	**F0**	**RS1**	**RS0**	**0V**	**-**	**P**

Serial Control (SCON)

Bit Addr.	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H
Name	**SM0**	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**

BIT	**SYMBOL**	**FUNCTION**
SCON.7	SM0	Serial port mode bit 0 (see Table A.1).
SCON.6	SM1	Serial port mode bit 1 (see Table A.1).
SCON.5	SM2	Serial port mode bit 2; enables multiprocessor communications in modes 2 and 3; RI will not be activated if received 9th bit is 0. In mode 1, if SM2 = 1, then RI will be activated only if a valid stop bit was received. In mode 0, SM2 should be 0.
SCON.4	REN	Receiver enable; must be set to receive characters.
SCON.3	TB8	Transmit bit 8; 9th bit transmitted in modes 2 and 3; set/cleared by software.
SCON.2	RB8	Receive bit 8; 9th bit received.
SCON.1	TI	Transmit interrupt flag; set at end of character transmission; cleared by software.
SCON.0	RI	Receive interrupt flag; set at end of character reception; cleared by software.

Table A.1 The 8051 Serial Port Mode Selection

SM0	**SM1**	**Mode**	**Description**	**Baud Rate**
0	0	0	Shift register	Fixed
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	Fixed
1	1	3	9-bit UART	Variable

PCON (Power Control)

Bit	7	6	5	4	3	2	1	0
Name	**SMOD**	**-**	**-**	**-**	**GF1**	**GF0**	**PD**	**IDL**

BIT	**SYMBOL**	**FUNCTION**
7	SMOD	Double the serial port baud rate when it is set to 1
3	GF1	General purpose flag bit 1
2	GF0	General purpose flag bit 0
1	PD	Activate the microcontroller power-down mode when it is set to 1
0	IDL	Activate the microcontroller idle mode when it is set to 1

Timer Control (TCON)

Bit Addr.	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H
Name	**TF1**	**TR1**	**TF0**	**TR0**	**IE1**	**IT1**	**IE0**	**IT0**

BIT	SYMBOL	FUNCTION
TCON.7	TF1	Timer-1 overflow flag. Set by hardware on overflow. Cleared by hardware when processor vectors to interrupt routine. Must be cleared by software when not involve interrupt
TCON.6	TR1	Timer-1 run control bit. Set/cleared by software to turn timer/counter on/off.
TCON.5	TF0	Timer-0 overflow flag. Do the same function as TF1 but for Timer-0.
TCON.4	TR0	Timer-0 run control bit. Do the same function as TR1 but for Timer-0.
TCON.3	IE1	External interrupt-1 edge flag. Set by hardware when interrupt-1 falling edge is detected. Cleared by hardware when interrupt is processed.
TCON.2	IT1	Interrupt-1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
TCON.1	IE0	External interrupt-1 edge flag. Do the same function as IE1 but for external interrupt-0.
TCON.0	IT0	Interrupt-0 Type control bit. Do the same function as IT1 but for external interrupt-0.

Timer Mode (TMOD)

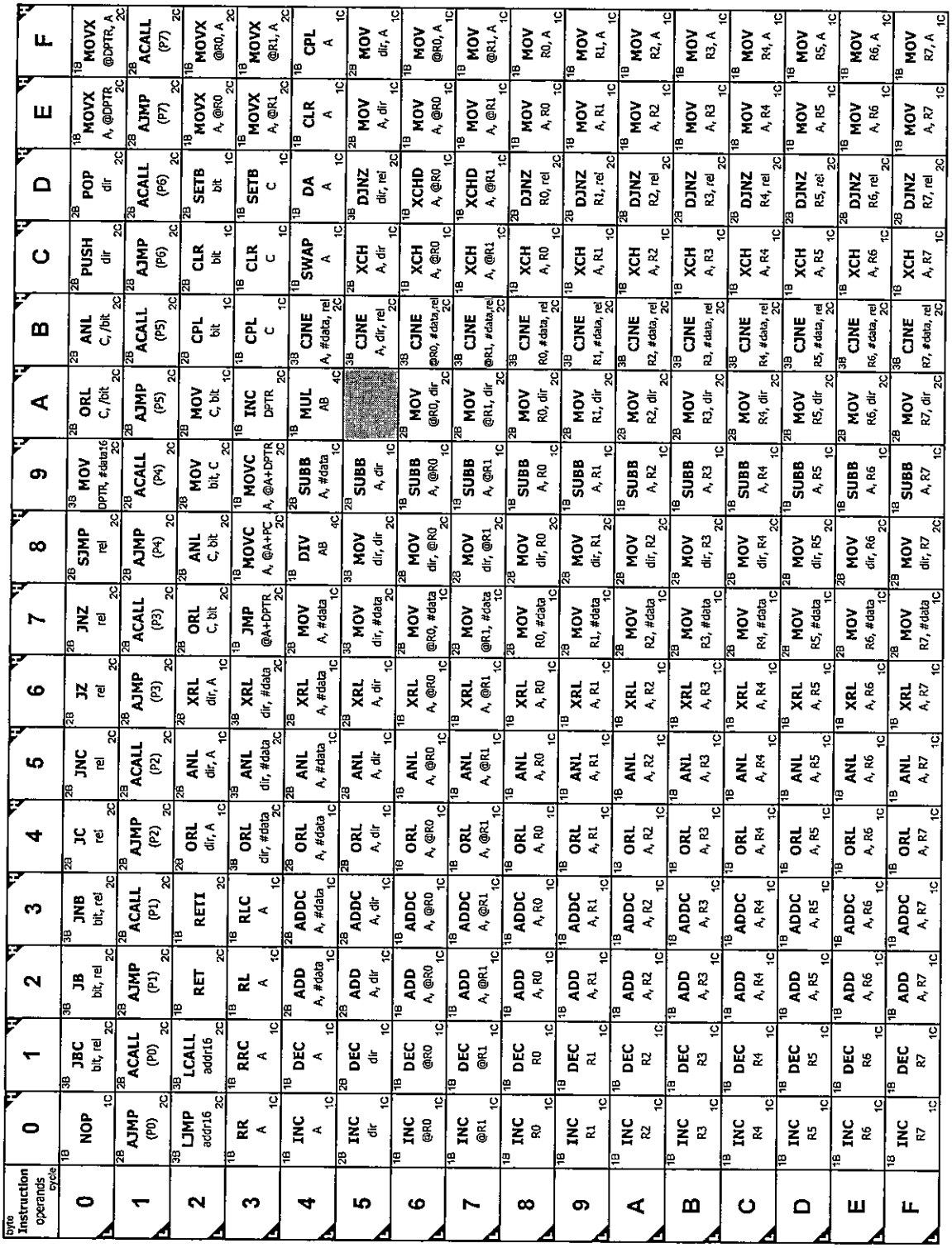
Bit	7	6	5	4	3	2	1	0
Name	**GATE**	**C/T**	**M1**	**M0**	**GATE**	**C/T**	**M1**	**M0**

BIT	SYMBOL	FUNCTION
TMOD.7	GATE1	When this bit is set the timer will only run when INT1 (P3.3) is high (hardware control). When this bit is cleared the timer will run regardless of the state of INT1 (software control).
TMOD.6	C/T1	Timer / Counter select bit. C / $\overline{T}$ = 0 → Timer operation. C / $\overline{T}$ = 1 → Counter operation.
TMOD.5	M1	Mode selection bits (see Table A.2). [for timer 1]
TMOD.4	M0	Mode selection bits (see Table A.2). [for timer 1]
TMOD.3	GATE0	Exactly the same function as GATE1 but for Timer0
TMOD.2	C/T0	Exactly the same function as C/T1 but for Timer0
TMOD.1	M1	Mode selection bits (see Table A.2). [for timer 0]
TMOD.0	M0	Mode selection bits (see Table A.2). [for timer 0]

Table A.2 Timer Mode Selection

M1	M0	Timer Mode	Description of Mode
0	0	0	13-bit Timer
0	1	1	16-bit Timer
1	0	2	8-bit auto-reload
1	1	3	Split timer mode

Opcode Map


byte
Instruction
operands
cycle
0 NOP 1C
1 JBC bit, rel 2C
2 JB bit, rel 2C
3 JNB bit, rel 2C
4 JC rel 2C
5 JNC rel 2C
6 JZ rel 2C
7 JNZ rel 2C
8 SJMP rel 2C
9 MOV DPTR, #data16 2C
A ORL C, /bit 2C
B ANL C, /bit 2C
C PUSH dir 2C
D POP dir 2C
E MOVX A, @DPTR 2C
F MOVX @DPTR, A 2C
0 AJMP (P0) 2C
1 ACALL (P0) 2C
2 AJMP (P1) 2C
3 ACALL (P1) 2C
4 AJMP (P2) 2C
5 ACALL (P2) 2C
6 AJMP (P3) 2C
7 ACALL (P3) 2C
8 AJMP (P4) 2C
9 ACALL (P4) 2C
A AJMP (P5) 2C
B ACALL (P5) 2C
C AJMP (P6) 2C
D ACALL (P6) 2C
E AJMP (P7) 2C
F ACALL (P7) 2C
0 LJMP addr16 2C
1 LCALL addr16 2C
2 RET 2C
3 RETI 2C
4 ORL dir, A 1C
5 ANL dir, A 1C
6 XRL dir, A 1C
7 ORL C, bit 2C
8 ANL C, bit 2C
9 MOV bit, C 2C
A MOV C, bit 1C
B CPL bit 1C
C CLR bit 1C
D SETB bit 1C
E MOVX A, @R0 2C
F MOVX @R0, A 2C
0 JBC bit, rel 2C
1 ACALL (P0) 2C
2 AJMP (P1) 2C
3 ACALL (P1) 2C
4 AJMP (P2) 2C
5 ACALL (P2) 2C
6 AJMP (P3) 2C
7 ACALL (P3) 2C
8 AJMP (P4) 2C
9 ACALL (P4) 2C
A AJMP (P5) 2C
B ACALL (P5) 2C
C AJMP (P6) 2C
D ACALL (P6) 2C
E AJMP (P7) 2C
F ACALL (P7) 2C
0 JBC bit, rel 2C
1 ACALL (P0) 2C
2 AJMP (P1) 2C
3 ACALL (P1) 2C
4 AJMP (P2) 2C
5 ACALL (P2) 2C
6 AJMP (P3) 2C
7 ACALL (P3) 2C
8 AJMP (P4) 2C
9 ACALL (P4) 2C
A AJMP (P5) 2C
B ACALL (P5) 2C
C AJMP (P6) 2C
D ACALL (P6) 2C
E AJMP (P7) 2C
F ACALL (P7) 2C
0 JBC bit, rel 2C
1 ACALL (P0) 2C
2 AJMP (P1) 2C
3 ACALL (P1) 2C
4 AJMP (P2) 2C
5 ACALL (P2) 2C
6 AJMP (P3) 2C
7 ACALL (P3) 2C
8 AJMP (P4) 2C
9 ACALL (P4) 2C
A AJMP (P5) 2C
B ACALL (P5) 2C
C AJMP (P6) 2C
D ACALL (P6) 2C
E AJMP (P7) 2C
F ACALL (P7) 2C
0 JBC bit, rel 2C
1 ACALL (P0) 2C
2 AJMP (P1) 2C
3 ACALL (P1) 2C
4 AJMP (P2) 2C
5 ACALL (P2) 2C
6 AJMP (P3) 2C
7 ACALL (P3) 2C
8 AJMP (P4) 2C
9 ACALL (P4) 2C
A AJMP (P5) 2C
B ACALL (P5) 2C
C AJMP (P6) 2C
D ACALL (P6) 2C
E AJMP (P7) 2C
F ACALL (P7) 2C
0 JBC bit, rel 2C
1 ACALL (P0) 2C
2 AJMP (P1) 2C
3 ACALL (P1) 2C
4 AJMP (P2) 2C
5 ACALL (P2) 2C
6 AJMP (P3) 2C
7 ACALL (P3) 2C
8 AJMP (P4) 2C
9 ACALL (P4) 2C
A AJMP (P5) 2C
B ACALL (P5) 2C
C AJMP (P6) 2C
D ACALL (P6) 2C
E AJMP (P7) 2C
F ACALL (P7) 2C
0 JBC bit, rel 2C
1 ACALL (P0) 2C
2 AJMP (P1) 2C
3 ACALL (P1) 2C
4 AJMP (P2) 2C
5 ACALL (P2) 2C
6 AJMP (P3) 2C
7 ACALL (P3) 2C
8 AJMP (P4) 2C
9 ACALL (P4) 2C
A AJMP (P5) 2C
B ACALL (P5) 2C
C AJMP (P6) 2C
D ACALL (P6) 2C
E AJMP (P7) 2C
F ACALL (P7) 2C
0 JBC bit, rel 2C
1 ACALL (P0) 2C
2 AJMP (P1) 2C
3 ACALL (P1) 2C
4 AJMP (P2) 2C
5 ACALL (P2) 2C
6 AJMP (P3) 2C
7 ACALL (P3) 2C
8 AJMP (P4) 2C
9 ACALL (P4) 2C
A AJMP (P5) 2C
B ACALL (P5) 2C
C AJMP (P6) 2C
D ACALL (P6) 2C
E AJMP (P7) 2C
F ACALL (P7) 2C
0 JBC bit, rel 2C
1 ACALL (P0) 2C
2 AJMP (P1) 2C
3 ACALL (P1) 2C
4 AJMP (P2) 2C
5 ACALL (P2) 2C
6 AJMP (P3) 2C
7 ACALL (P3) 2C
8 AJMP (P4) 2C
9 ACALL (P4) 2C
A AJMP (P5) 2C
B ACALL (P5) 2C
C AJMP (P6) 2C
D ACALL (P6) 2C
E AJMP (P7) 2C
F ACALL (P7) 2C
0 JBC bit, rel 2C
1 ACALL (P0) 2C
2 AJMP (P1) 2C
3 ACALL (P1) 2C
4 AJMP (P2) 2C
5 ACALL (P2) 2C
6 AJMP (P3) 2C
7 ACALL (P3) 2C
8 AJMP (P4) 2C
9 ACALL (P4) 2C
A AJMP (P5) 2C
B ACALL (P5) 2C
C AJMP (P6) 2C
D ACALL (P6) 2C
E AJMP (P7) 2C
F ACALL (P7) 2C